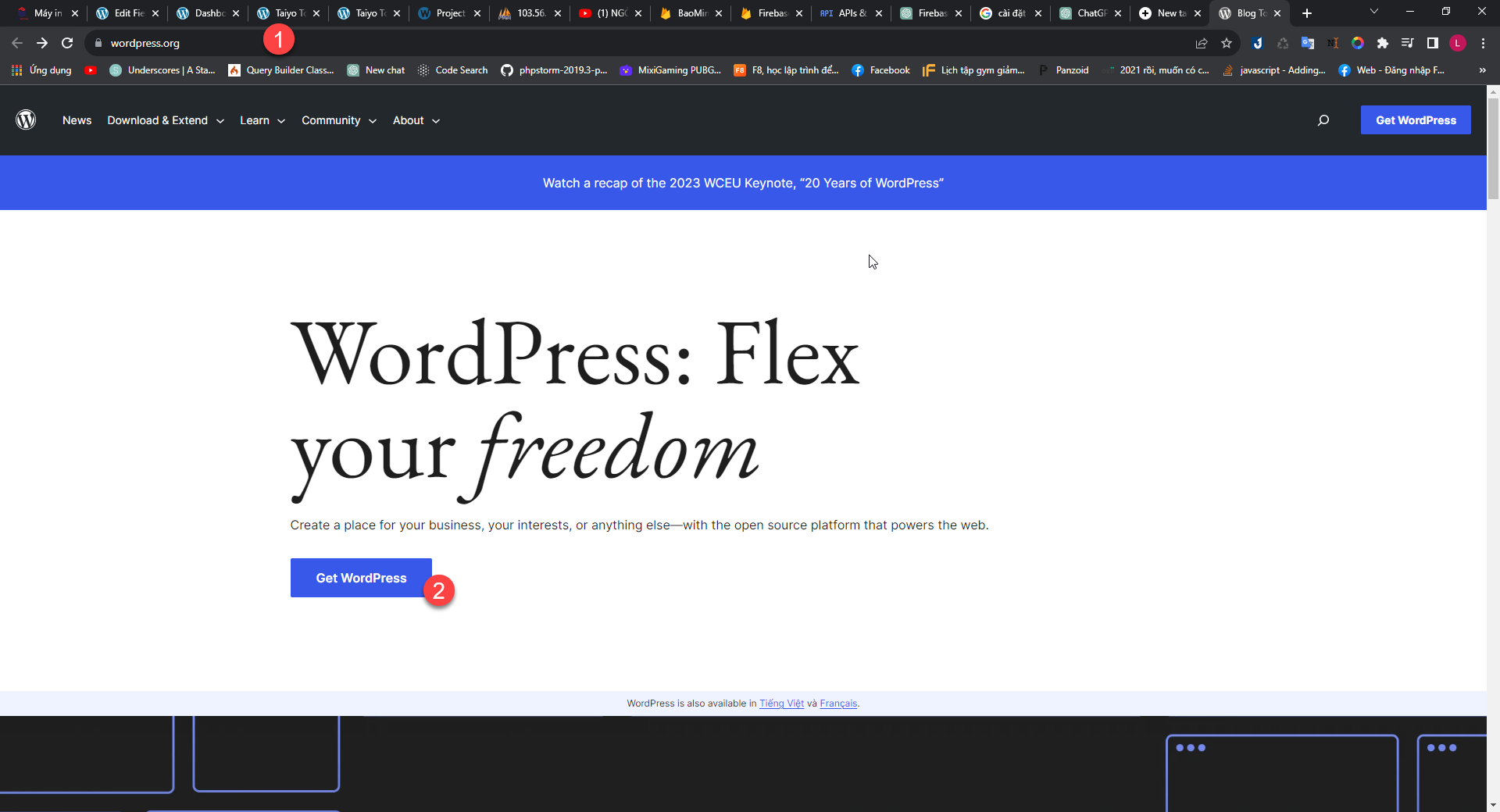
# Hướng dẫn cơ bản wordpress

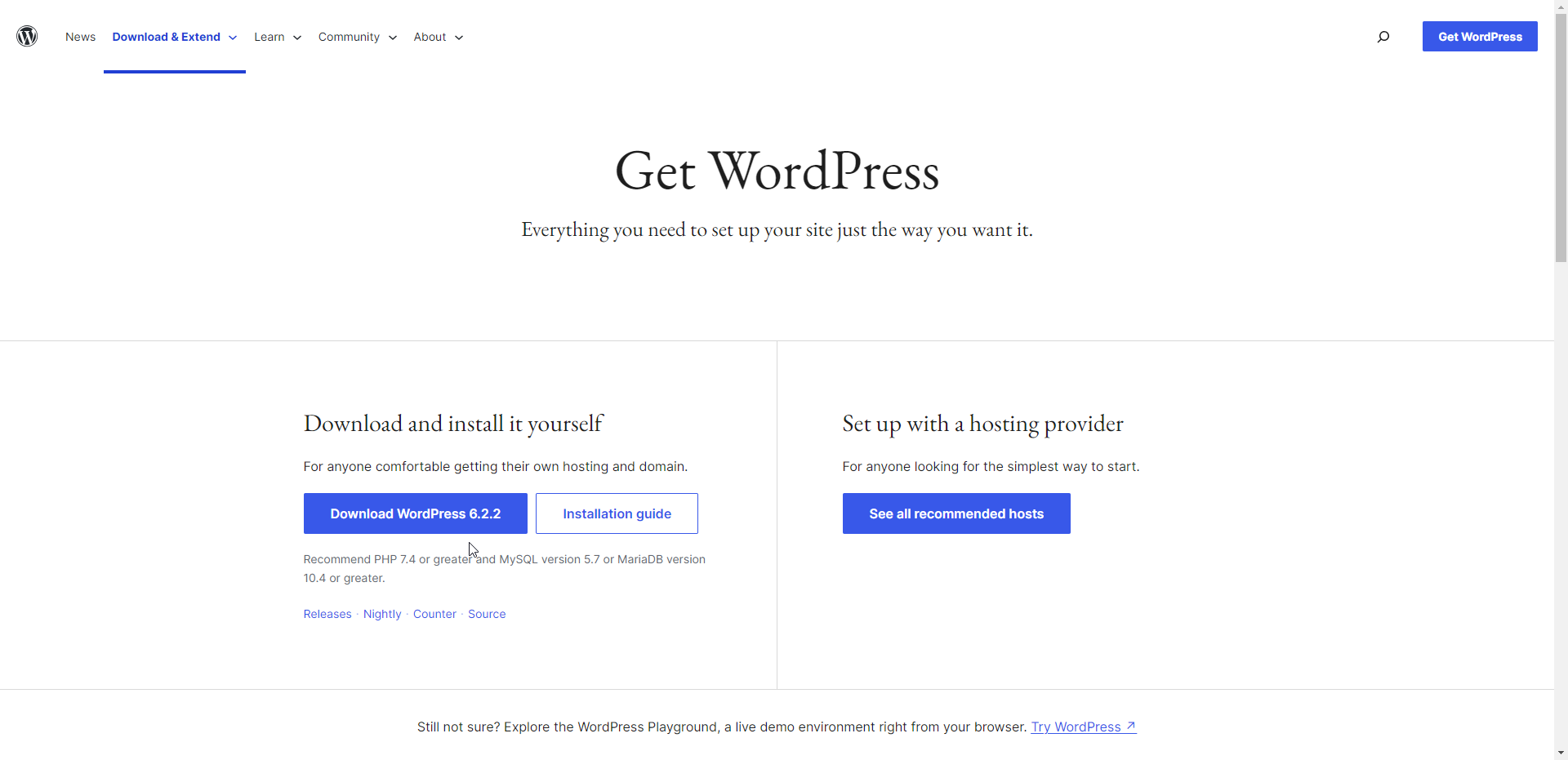
1. Cài đặt WordPress
   1. Giới thiệu sơ lược

WordPress là một hệ thống quản lý nội dung mã nguồn mở phổ biến, cho phép bạn tạo và quản lý các trang web và blog một cách dễ dàng. Với hàng ngàn chủ đề và plugin, bạn có thể tùy chỉnh giao diện và chức năng của trang web theo ý muốn.

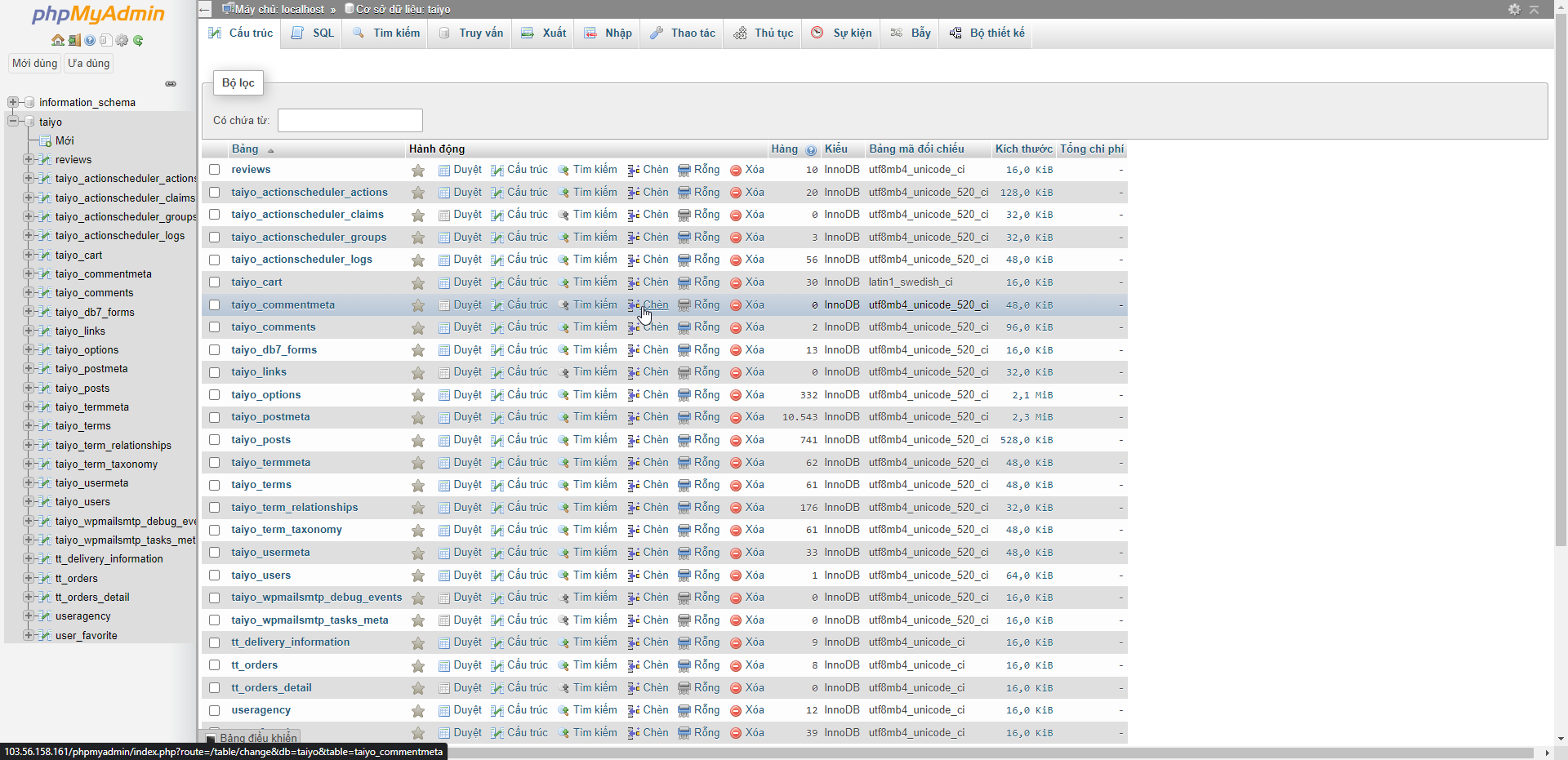
* 1. Cài đặt WordPress
     1. Tải và giải nén tập tin
        1. Truy cập trang chính thức của WordPress tại [https://wordpress.org](https://wordpress.org/).
        2. Nhấp vào nút "Get WordPress" để tải xuống tệp nén ZIP của WordPress.



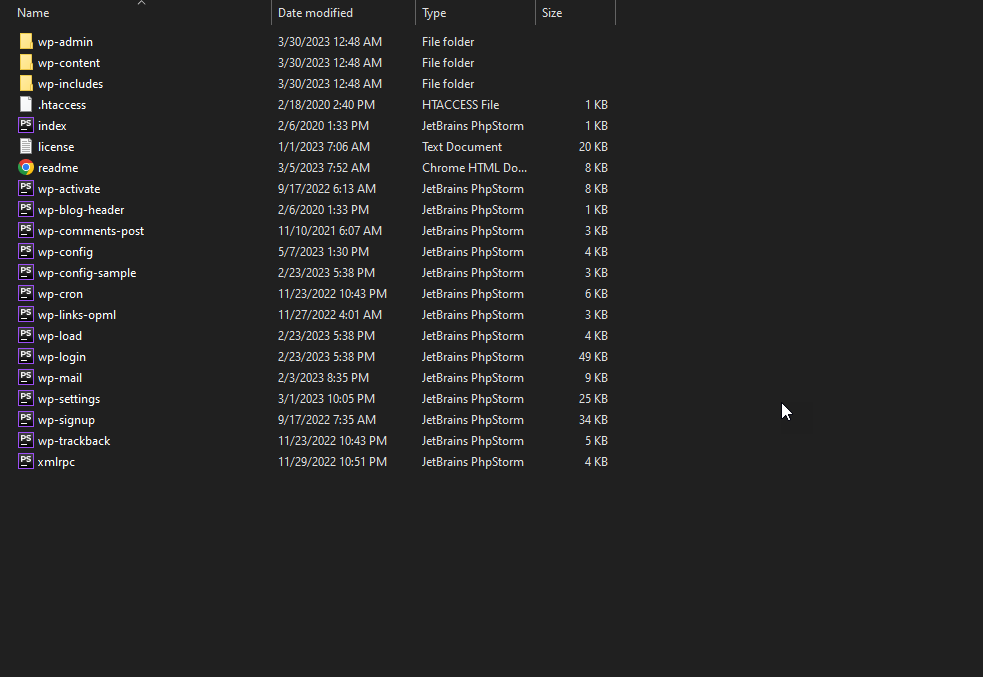
* + - 1. Mở tệp ZIP sau khi tải xong và giải nén nó vào thư mục trên máy tính của bạn.



* + 1. Tạo cơ sở dữ liệu
       1. Đăng nhập vào hệ thống quản lý cơ sở dữ liệu của bạn (thông qua phpMyAdmin hoặc công cụ tương tự). - Ở đây sử dụng phpMyAdmin



* + - 1. Tạo một cơ sở dữ liệu mới cho trang web WordPress của bạn. Ghi nhớ tên cơ sở dữ liệu, tên người dùng và mật khẩu.
    1. Cấu hình tệp wp-config.php
       1. Trong thư mục giải nén của WordPress, tìm và mở tệp **wp-config-sample.php** bằng trình soạn thảo văn bản.



* + - 1. Điền thông tin cơ sở dữ liệu mà bạn đã tạo ở bước trước vào các phần:
         1. DB\_NAME: Tên cơ sở dữ liệu
         2. DB\_USER: Tên người dùng cơ sở dữ liệu
         3. DB\_PASSWORD: Mật khẩu cơ sở dữ liệu
      2. Lưu tệp với tên **wp-config.php**.
    1. Hoàn tất quá trình cài đặt
       1. Tải lên toàn bộ thư mục giải nén WordPress vào thư mục gốc của máy chủ web của bạn, sử dụng FTP hoặc công cụ quản lý tệp của nhà cung cấp hosting.
       2. Truy cập trang web của bạn qua trình duyệt web. Bạn sẽ thấy trang hướng dẫn hoàn tất quá trình cài đặt.
       3. Điền thông tin về trang web, tạo tài khoản quản trị viên và sau đó nhấn "Install WordPress".
       4. Quá trình cài đặt sẽ diễn ra trong vài giây. Sau khi hoàn tất, bạn có thể đăng nhập vào trang quản trị WordPress và bắt đầu tùy chỉnh và thêm nội dung vào trang web của mình.
       5. Hy vọng rằng hướng dẫn này sẽ giúp bạn cài đặt WordPress một cách thành công và bắt đầu xây dựng trang web của mình.

1. Chức năng của các thư mục trong WordPress
   1. wp-content

Thư mục wp-content chứa tất cả các dữ liệu tùy chỉnh và tương tác của bạn với WordPress. Đây là nơi bạn tạo ra và lưu trữ các nội dung tùy chỉnh như chủ đề, plugin, hình ảnh, video và các tệp tương tự. Cụ thể:

* + 1. themes: Thư mục này chứa các chủ đề (themes) mà bạn có thể cài đặt để thay đổi giao diện của trang web.
    2. plugins: Chứa các plugin mở rộng chức năng của WordPress. Bạn có thể cài đặt các plugin để thêm tính năng mới cho trang web.
    3. uploads: Chứa các tệp được tải lên bởi người dùng, bao gồm hình ảnh, video, tệp tin tải lên thông qua trang quản lý phương tiện của WordPress.
  1. wp-admin
     1. Thư mục wp-admin chứa tất cả các tệp liên quan đến giao diện quản trị của trang web WordPress. Đây là nơi bạn thực hiện các hoạt động quản lý trang web, chỉnh sửa nội dung và cài đặt chung. Giao diện quản trị (Admin Dashboard) của WordPress được truy cập thông qua thư mục này. Các vai trò chính của wp-admin bao gồm:
     2. Quản lý: Tạo, chỉnh sửa và xóa bài viết, trang, danh mục, tùy chỉnh giao diện, cài đặt plugin, và nhiều hoạt động quản lý khác.
     3. Cấu hình: Tùy chỉnh các thiết lập chung của trang web, bao gồm cài đặt chung, cài đặt định dạng bài viết, cài đặt đa phương tiện và nhiều tùy chọn khác.
     4. Quản lý người dùng: Tạo và quản lý người dùng, gán vai trò, quyền truy cập và quản lý bình luận.
  2. wp-includes
     1. Thư mục wp-includes chứa các tệp mã nguồn chính của WordPress, bao gồm các thư viện và tệp hỗ trợ cần thiết để hoạt động của WordPress. Thư mục này không nên bị chỉnh sửa, vì những thay đổi có thể gây ra lỗi cho hệ thống. Một số vai trò của wp-includes bao gồm:
     2. Thư viện: Chứa mã nguồn chính của WordPress, bao gồm các hàm và lớp được sử dụng bởi các chức năng khác trong trang web.
     3. Tệp hỗ trợ: Chứa các tệp dùng để quản lý các chức năng như xử lý dữ liệu, quản lý định dạng nội dung, quản lý đa phương tiện và nhiều chức năng hệ thống khác.
     4. Tóm lại, các thư mục chính wp-content, wp-admin, và wp-includes có vai trò quan trọng trong việc xây dựng, quản lý và hiển thị trang web WordPress của bạn. Chúng là các phần cơ bản tạo nên hệ thống quản lý nội dung mạnh mẽ và linh hoạt.
  3. Những lưu ý:

khi sử dụng ftp đẩy lên host, phải thay đổi permission thành 777.

Thư mục wp-admin và wp-includes không được phép sửa đổi trong đó

1. Tạo theme WordPress
   1. Tạo một chủ đề WordPress tùy chỉnh có thể được thực hiện bằng cách sử dụng một trình tạo chủ đề như Underscores (\_s). Dưới đây là hướng dẫn cơ bản để tạo chủ đề bằng Underscores:
      1. Truy cập trang web Underscores: https://underscores.me/
      2. Điền tên chủ đề vào ô "Theme Name". Đây là tên chủ đề bạn muốn tạo.
      3. Chọn các tùy chọn khác như "Author Name" và "Author URL" (tùy chọn).
      4. Chọn các tính năng bạn muốn bao gồm trong chủ đề. Điều này có thể bao gồm hỗ trợ cho hình ảnh đặt nền, hỗ trợ cho giao diện di động, hỗ trợ cho tùy chọn chủ đề thêm, v.v.
      5. Nhấn nút "Generate" để tạo chủ đề. Sau khi tạo xong, bạn sẽ có tùy chọn tải chủ đề về máy tính dưới dạng tệp nén ZIP.
   2. Sau khi tải chủ đề về máy tính, bạn có thể tiến hành cài đặt và tùy chỉnh chủ đề theo mong muốn:
      1. Giải nén tệp nén ZIP bạn vừa tải về.
      2. Sao chép thư mục chứa tên chủ đề mới được tạo (ví dụ: my\_custom\_theme) vào thư mục wp-content/themes trong thư mục gốc của trang web WordPress.
      3. Trong trang quản trị WordPress, truy cập mục "Themes" trong menu "Appearance".
      4. Kích hoạt chủ đề bạn vừa tạo lên trang web của bạn.
      5. Từ đó, bạn có thể bắt đầu tùy chỉnh chủ đề bằng cách sửa các tệp mã nguồn và tệp mẫu (template files) trong thư mục chủ đề. Điều này cho phép bạn thay đổi giao diện, chức năng và phong cách của trang web theo ý muốn.

Lưu ý rằng tạo và tùy chỉnh chủ đề yêu cầu kiến thức về lập trình WordPress, HTML, CSS và PHP. Underscores (\_s) cung cấp một kiến trúc cơ bản để bạn có thể tạo chủ đề theo ý muốn.

1. Cấu trúc tệp theme cơ bản của Wordpress

Cấu trúc tệp trong một chủ đề WordPress được định dạng theo các thư mục và tệp chuẩn để quản lý giao diện và chức năng của trang web. Dưới đây là cấu trúc cơ bản của thư mục chủ đề trong WordPress:

my\_custom\_theme/ # Thư mục chủ đề gốc

|-- style.css # Tệp CSS chính của chủ đề

|-- index.php # Tệp mã nguồn chính hiển thị trang chủ

|-- header.php # Tệp mã nguồn đầu trang (header)

|-- footer.php # Tệp mã nguồn chân trang (footer)

|-- functions.php # Tệp chức năng chủ đề

|-- single.php # Tệp mã nguồn hiển thị bài viết đơn

|-- page.php # Tệp mã nguồn hiển thị trang tĩnh

|-- archive.php # Tệp mã nguồn hiển thị các trang lưu trữ

|-- search.php # Tệp mã nguồn hiển thị kết quả tìm kiếm

|-- 404.php # Tệp mã nguồn hiển thị trang 404

|-- category.php # Tệp mã nguồn hiển thị danh mục

|-- tag.php # Tệp mã nguồn hiển thị tag

|-- comments.php # Tệp mã nguồn hiển thị phần bình luận

|-- content.php # Tệp mã nguồn hiển thị nội dung bài viết

|-- sidebar.php # Tệp mã nguồn hiển thị sidebar

|-- searchform.php # Tệp mã nguồn hiển thị biểu mẫu tìm kiếm

|-- category.php # Tệp mã nguồn hiển thị danh mục

|-- tag.php # Tệp mã nguồn hiển thị tag

|-- taxonomy-{tên\_slug}.php # Tệp mã nguồn hiển thị danh mục cụ thể

|-- images/ # Thư mục chứa hình ảnh tùy chỉnh

|-- js/ # Thư mục chứa các tệp JavaScript

|-- css/ # Thư mục chứa các tệp CSS tùy chỉnh

|-- templates/ # Thư mục chứa các tệp mẫu tùy chỉnh

| |-- custom-template.php # Tệp mẫu tùy chỉnh

Dưới đây là mô tả của một số tệp quan trọng trong cấu trúc chủ đề:

* 1. style.css: Tệp CSS chính của chủ đề. Nó chứa thông tin về chủ đề như tên, tác giả, phiên bản, mô tả và các ghi chú khác.
  2. index.php: Tệp mã nguồn chính được gọi khi hiển thị trang chủ.
  3. header.php: Mã nguồn hiển thị đầu trang chung cho mọi trang.
  4. footer.php: Mã nguồn hiển thị chân trang chung cho mọi trang.
  5. functions.php: Tệp chứa các hàm và chức năng tùy chỉnh của chủ đề.
  6. single.php: Mã nguồn hiển thị cho bài viết đơn.
  7. page.php: Mã nguồn hiển thị cho các trang tĩnh.
  8. archive.php: Mã nguồn hiển thị cho các trang lưu trữ (ví dụ: danh mục, tag).
  9. search.php: Mã nguồn hiển thị kết quả tìm kiếm.
  10. 404.php: Mã nguồn hiển thị trang 404 (không tìm thấy).
  11. comments.php: Mã nguồn hiển thị phần bình luận.
  12. sidebar.php: Mã nguồn hiển thị sidebar.
  13. searchform.php: Mã nguồn hiển thị biểu mẫu tìm kiếm.
  14. images/: Thư mục chứa hình ảnh tùy chỉnh của chủ đề.
  15. js/: Thư mục chứa các tệp JavaScript.
  16. css/: Thư mục chứa các tệp CSS tùy chỉnh.
  17. templates/: Thư mục chứa các tệp mẫu tùy chỉnh. Đây là nơi bạn có thể tạo các tệp mẫu riêng cho các trang hoặc loại trang cụ thể.
  18. taxonomy-{tên\_slug}.php: Tệp mã nguồn hiển thị danh mục cụ thể. Thay thế {tên\_slug} bằng slug thực tế của danh mục bạn muốn tạo tệp mẫu.

Với thư mục templates, bạn có thể tạo các tệp mẫu tùy chỉnh cho các trang hoặc phần cụ thể của trang web của bạn. Và với tệp taxonomy-{tên\_slug}.php, bạn có thể tùy chỉnh cách hiển thị cho các danh mục cụ thể trong trang web.

Khi bạn tạo chủ đề mới, bạn có thể sửa các tệp này để tạo giao diện và chức năng theo ý muốn. Hãy nhớ rằng việc tạo chủ đề đòi hỏi hiểu biết về HTML, CSS và PHP.

1. Lập trình code tạo template page, single-post, category, taxonomy, tag, header, footer,..
   1. Header.php, footer.php

Nguồn tham khảo header: <https://github.com/WordPress/WordPress/blob/master/wp-content/themes/twentyfourteen/header.php>

Nguồn tham khảo footer: <https://github.com/WordPress/WordPress/blob/master/wp-content/themes/twentyfourteen/footer.php>

Những lưu ý:

Các hàm thường sử dụng.

* wp\_head(): Hàm này thường được đặt trong thẻ <head> của tệp header.php. Nó cho phép các plugin và các phần khác của WordPress chèn các mã và tài nguyên vào phần đầu trang của trang web, chẳng hạn như các tệp CSS, JavaScript, và các meta tag quan trọng.
* bloginfo('name'): Hàm này hiển thị tên của trang web.
* bloginfo('description'): Hàm này hiển thị mô tả của trang web.
* wp\_nav\_menu(): Hàm này được sử dụng để hiển thị menu của trang web.
* wp\_footer(): Hàm này thường được đặt trước thẻ đóng của thẻ <body> trong tệp footer.php. Tương tự như wp\_head(), hàm này cho phép các plugin và các phần khác của WordPress chèn các mã và tài nguyên vào phần chân trang của trang web. get\_sidebar(): Hàm này được sử dụng để hiển thị sidebar (cột bên) của trang web.
* dynamic\_sidebar(): Hàm này được sử dụng để hiển thị các widget được thêm vào sidebar.
* wp\_nav\_menu(): Hàm này cũng có thể được sử dụng trong footer.php để hiển thị menu.
* bloginfo('name'): Hàm này cũng có thể được sử dụng để hiển thị tên của trang web.
* bloginfo('description'): Hàm này cũng có thể được sử dụng để hiển thị mô tả của trang web.

Trên đây chỉ là một số hàm phổ biến và thường được sử dụng trong các tệp header.php và footer.php trong WordPress. Bạn cũng có thể tạo các mã HTML và CSS tùy chỉnh để thiết kế giao diện phù hợp với trang web của bạn.

* 1. page, single-post
     1. get\_header() và get\_footer(): Được sử dụng để lấy và hiển thị phần đầu trang (header.php) và phần chân trang (footer.php) của trang web.
     2. have\_posts() và the\_post(): Sử dụng để kiểm tra xem có bài viết nào trong vòng lặp và hiển thị bài viết tiếp theo.
     3. the\_title(): Hiển thị tiêu đề của bài viết hoặc trang.
     4. the\_content(): Hiển thị nội dung của bài viết hoặc trang.
     5. the\_excerpt(): Hiển thị một phần tóm tắt của nội dung của bài viết.
     6. the\_permalink(): Trả về URL của bài viết hoặc trang hiện tại.
     7. the\_category(): Hiển thị danh mục của bài viết.
     8. the\_tags(): Hiển thị các thẻ của bài viết.
     9. the\_author(): Hiển thị tên tác giả của bài viết.
     10. get\_the\_date(): Trả về ngày viết bài.
     11. get\_the\_post\_thumbnail(): Lấy hình ảnh đại diện của bài viết.
     12. wp\_list\_categories(): Liệt kê danh mục.
  2. category, taxonomy, tag
     1. get\_terms(): Lấy danh sách các thuộc tính của một loại bài viết như danh mục hoặc thẻ.
     2. is\_category(), is\_tag(), is\_tax(): Kiểm tra xem trang hiện tại có phải là trang danh mục, thẻ, hoặc thuộc tính không.
     3. get\_queried\_object(): Trả về đối tượng hiện đang được truy vấn (danh mục, thẻ, hoặc thuộc tính).
     4. get\_posts(): Lấy danh sách các bài viết dựa trên các tham số truy vấn.

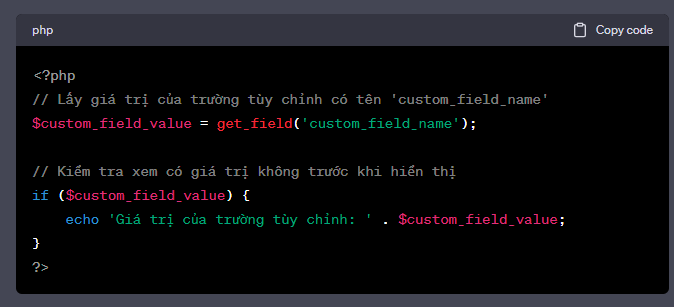
Ngoài ra, còn có rất nhiều hàm có thể áp dụng, có thể tham khảo hoặc tìm kiếm trên : <https://codex.wordpress.org/Function_Reference>

1. Các plugin hay sử dụng
   1. Advanced Custom Fields (ACF)
      1. Giới thiệu

Advanced Custom Fields (ACF) là một plugin mạnh mẽ cho WordPress cho phép bạn tạo các trường tùy chỉnh trong bài viết, trang và các loại nội dung khác. Điều này cho phép bạn tùy chỉnh nội dung và dữ liệu hiển thị trên trang web của mình một cách linh hoạt và dễ dàng..

* + 1. Cài đặt cà kích hoạt ACF
       1. Đăng nhập vào trang quản trị WordPress của bạn.
       2. Trong menu bên trái, di chuột qua "Plugins" và chọn "Add New".
       3. Tìm kiếm "Advanced Custom Fields" và cài đặt plugin có tên giống như vậy.
       4. Kích hoạt plugin sau khi cài đặt.
    2. Tạo trường tùy chỉnh:
       1. Sau khi kích hoạt, trong menu bên trái, bạn sẽ thấy một mục mới có tên "Custom Fields". Di chuột qua nó và chọn "Add New".
       2. Tại đây, bạn có thể tạo các trường tùy chỉnh mới. Chọn loại trường mà bạn muốn (ví dụ: text, số, hình ảnh, URL, v.v.) và cấu hình các tùy chọn cho trường đó.
    3. Gắn trường tùy chỉnh vào bài viết hoặc trang:
       1. Khi bạn tạo hoặc chỉnh sửa một bài viết hoặc trang, bạn sẽ thấy các trường tùy chỉnh mà bạn đã tạo xuất hiện dưới nội dung bài viết hoặc trang.
       2. Điền thông tin vào các trường tùy chỉnh theo ý muốn.
    4. Lấy giá trị của trường tùy chỉnh trong mã PHP:
       1. Bạn có thể sử dụng hàm get\_field() hoặc the\_field() để lấy giá trị của các trường tùy chỉnh trong mã PHP.
       2. Ví dụ: Nếu bạn tạo một trường tùy chỉnh có tên là "custom\_field\_name", bạn có thể sử dụng get\_field('custom\_field\_name') để lấy giá trị của trường đó.

Dưới đây là ví dụ cách sử dụng hàm get\_field() để lấy giá trị của một trường tùy chỉnh và hiển thị nó trong tệp mẫu single.php:



* 1. Contact Form 7 và SMTP
     1. Giới thiệu

**Contact Form 7** là một plugin cho WordPress cho phép bạn dễ dàng tạo các biểu mẫu liên hệ và quản lý chúng trên trang web của bạn.

**Easy WP SMTP** là một plugin khác giúp bạn cấu hình máy chủ SMTP để gửi email từ WordPress, đảm bảo email của bạn được gửi đúng cách.

* + 1. Cài đặt và cấu hình Contact Form 7:
       1. Đăng nhập vào trang quản trị WordPress của bạn.
       2. Trong menu bên trái, di chuột qua "Plugins" và chọn "Add New".
       3. Tìm kiếm "Contact Form 7" và cài đặt plugin có tên giống như vậy.
       4. Kích hoạt plugin sau khi cài đặt.
       5. Trong menu bên trái, bạn sẽ thấy mục mới có tên "Contact" xuất hiện. Di chuột qua nó và chọn "Add New".
       6. Tại đây, bạn có thể tạo biểu mẫu liên hệ theo ý muốn bằng cách sử dụng các thẻ shortcode. Cấu hình các trường, thông điệp cảm ơn, và cài đặt khác cho biểu mẫu.
    2. Cài đặt và cấu hình Easy WP SMTP:
       1. rong menu bên trái, di chuột qua "Plugins" và chọn "Add New".
       2. Tìm kiếm "Easy WP SMTP" và cài đặt plugin có tên giống như vậy.
       3. Kích hoạt plugin sau khi cài đặt.
       4. Trong menu bên trái, bạn sẽ thấy mục "Easy WP SMTP". Nhấp vào đó để cấu hình.
    3. Cấu hình Easy WP SMTP để gửi email:
       1. Trong trang cấu hình của Easy WP SMTP, bạn cần điền thông tin máy chủ SMTP của bạn (thường do nhà cung cấp hosting cung cấp), tên người dùng và mật khẩu.
       2. Bạn cũng có thể cấu hình các tùy chọn bổ sung như gửi email thông qua SSL hoặc TLS.
       3. Lưu cấu hình sau khi hoàn thành.
    4. Tạo biểu mẫu liên hệ với Contact Form 7:
       1. Sau khi cài đặt và cấu hình Contact Form 7, bạn có thể tạo biểu mẫu liên hệ bằng cách tạo một trang hoặc bài viết mới.
       2. Sử dụng các thẻ shortcode mà bạn đã cấu hình trong biểu mẫu Contact Form 7 để hiển thị biểu mẫu trên trang.
    5. Kết nối biểu mẫu liên hệ với Easy WP SMTP:
       1. Trong biểu mẫu liên hệ của bạn trong Contact Form 7, bạn có thể chỉ định địa chỉ email mà bạn muốn nhận thông điệp từ biểu mẫu.
       2. Để đảm bảo email được gửi thông qua máy chủ SMTP, bạn không cần phải thay đổi bất kỳ cài đặt gì trong biểu mẫu liên hệ. Plugin Easy WP SMTP sẽ xử lý việc gửi email.

Lưu ý rằng cấu hình máy chủ SMTP có thể thay đổi tùy theo nhà cung cấp hosting của bạn. Đảm bảo bạn có thông tin chính xác từ nhà cung cấp hosting hoặc nhà cung cấp dịch vụ email của bạn.

* 1. custom post type (CPT/UI)
     1. Giới thiệu

Tạo một Custom Post Type (CPT) trong WordPress cho phép bạn tạo các loại nội dung tùy chỉnh như sản phẩm, dự án, sự kiện, v.v. Dưới đây là cách tạo Custom Post Type sử dụng Custom Post Type UI plugin:

* + 1. Cài đặt và cấu hình Custom Post Type UI
       1. Đăng nhập vào trang quản trị WordPress của bạn.
       2. Trong menu bên trái, di chuột qua "Plugins" và chọn "Add New".
       3. Tìm kiếm "Custom Post Type UI" và cài đặt plugin có tên giống như vậy.
       4. Kích hoạt plugin sau khi cài đặt.
       5. Trong menu bên trái, bạn sẽ thấy mục "CPT UI" xuất hiện. Di chuột qua nó và chọn "Add/Edit Post Types".
    2. Tạo Custom Post Type
       1. Trong trang "CPT UI", bạn có thể tạo mới một Custom Post Type bằng cách điền các thông tin sau:
          1. Tên của Custom Post Type.
          2. Tùy chọn danh mục (nếu cần).
          3. Thiết lập giao diện và quản lý nội dung (nếu cần).
          4. Tùy chọn hỗ trợ cho các tính năng như "mục lục", "nhãn", "hình đặc trưng", v.v.
       2. Sau khi hoàn thành, nhấn "Add Post Type" để tạo Custom Post Type mới.
    3. Tạo các mẫu trang cho Custom Post Type:
       1. Sau khi tạo Custom Post Type, bạn cần tạo các mẫu trang để hiển thị nội dung của Custom Post Type.
       2. Bạn có thể sử dụng các tệp mẫu như single-{post\_type}.php hoặc archive-{post\_type}.php để tạo giao diện hiển thị nội dung cho từng bài viết hoặc danh sách bài viết của Custom Post Type.
    4. Thêm dữ liệu vào Custom Post Type
       1. Sau khi tạo Custom Post Type và tạo mẫu trang, bạn có thể thêm nội dung mới thông qua trang quản lý trong WordPress.
       2. Trong trang quản lý, bạn sẽ thấy mục mới tương ứng với Custom Post Type mà bạn đã tạo. Bạn có thể thêm, sửa và xóa các mục nội dung trong đó.
    5. Hiển thị dữ liệu trong tệp mẫu:
       1. Sử dụng các hàm WordPress như have\_posts(), the\_post(), và các hàm khác để hiển thị nội dung của Custom Post Type trong các tệp mẫu bạn đã tạo.
       2. Bạn cũng có thể sử dụng hàm get\_post\_meta() để lấy dữ liệu tùy chỉnh từ các trường tùy chỉnh của Custom Post Type.

Nhớ rằng, Custom Post Type UI chỉ là một trong nhiều cách để tạo Custom Post Type. Bạn cũng có thể tạo Custom Post Type bằng cách viết mã nguồn trực tiếp vào tệp functions.php của chủ đề hoặc sử dụng các plugin tạo chủ đề khác nhau.

* 1. Classic Editor
     1. Giới thiệu

Plugin Classic Editor cho phép bạn sử dụng giao diện soạn thảo "Classic Editor" trong WordPress thay vì giao diện soạn thảo Gutenberg (Block Editor). Đây là một giải pháp cho những người thích sử dụng giao diện soạn thảo cũ hoặc muốn tiếp tục sử dụng các plugin được tạo dựa trên giao diện cũ. Dưới đây là cách sử dụng plugin Classic Editor:

* + 1. Cài đặt và kích hoạt Classic Editor
       1. Đăng nhập vào trang quản trị WordPress của bạn.
       2. Trong menu bên trái, di chuột qua "Plugins" và chọn "Add New".
       3. Tìm kiếm "Classic Editor" và cài đặt plugin có tên giống như vậy.
       4. Kích hoạt plugin sau khi cài đặt.
    2. Cấu hình plugin Classic Editor
       1. Sau khi kích hoạt, bạn có thể cấu hình cách plugin sẽ hoạt động.
       2. Trong trang cài đặt, bạn có thể chọn giữ giao diện "Classic Editor" cho tất cả người dùng hoặc để mỗi người dùng tự chọn giao diện
    3. Sử dụng Classic Editor
       1. Sau khi cấu hình, bạn sẽ thấy rằng giao diện soạn thảo cho bài viết và trang sẽ là giao diện "Classic Editor" thay vì Gutenberg.
       2. Giao diện này cho phép bạn sử dụng trình soạn thảo dạng văn bản thông thường mà bạn đã quen thuộc.
    4. Tùy chỉnh bài viết và trang
       1. Khi bạn tạo hoặc chỉnh sửa bài viết hoặc trang, bạn sẽ sử dụng giao diện "Classic Editor" để thêm và định dạng nội dung.
       2. Bạn cũng có thể sử dụng các plugin được thiết kế dựa trên giao diện cũ mà bạn đã quen thuộc.

function display\_other\_slider($atts) {

    ob\_start(); // Bắt đầu bộ đệm đầu ra

    $other\_sliders = new WP\_Query(array(

        'post\_type' => 'other\_slider',

        'posts\_per\_page' => -1

    ));

    if ($other\_sliders->have\_posts()) :

    ?>

        <div class="other">

        <div class="other-slider">

            <?php

            while ($other\_sliders->have\_posts()) : $other\_sliders->the\_post();

                $other\_slider\_items = get\_field('knowledge\_slider'); // Đổi thành tên trường Repeater của bạn

                if ($other\_slider\_items) :

                    foreach ($other\_slider\_items as $item) :

                        $image = $item['knowledge\_slider\_image'];

                        $text = $item['knowledge\_slider\_text'];

            ?>

                        <div class="child">

                            <div class="image">

                                <figure>

                                    <img src="<?php echo esc\_url($image['url']); ?>" alt="<?php echo esc\_attr($image['alt']); ?>">

                                </figure>

                            </div>

                            <div class="text">

                                <h4><?php echo esc\_html($text); ?></h4>

                            </div>

                        </div>

            <?php

                    endforeach;

                endif;

            endwhile;

            ?>

        </div>

        </div>

    <?php

        wp\_reset\_postdata();

    endif;

    return ob\_get\_clean(); // Kết thúc bộ đệm đầu ra và trả về dữ liệu đã hiển thị

}

add\_shortcode('display\_other\_slider', 'display\_other\_slider');